



Cable, H., & Browne, D. E. (2015). Exact and efficient simulation of concordant computation. *New Journal of Physics*, 17(11), [113049]. <https://doi.org/10.1088/1367-2630/17/11/113049>

Publisher's PDF, also known as Version of record

Link to published version (if available):
[10.1088/1367-2630/17/11/113049](https://doi.org/10.1088/1367-2630/17/11/113049)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

Exact and efficient simulation of concordant computation

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2015 New J. Phys. 17 113049

(<http://iopscience.iop.org/1367-2630/17/11/113049>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 137.222.10.113

This content was downloaded on 22/11/2015 at 21:33

Please note that [terms and conditions apply](#).



PAPER

Exact and efficient simulation of concordant computation

Hugo Cable^{1,3} and Daniel E Browne²¹ Centre for Quantum Photonics, H. H. Wills Physics Laboratory and Department of Electrical and Electronic Engineering, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK² Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, UK³ Author to whom any correspondence should be addressed.E-mail: Hugo.Cable@bristol.ac.uk and d.browne@ucl.ac.uk

Keywords: quantum computing, quantum algorithms, quantum discord

RECEIVED
8 September 2015REVISED
26 October 2015ACCEPTED FOR PUBLICATION
27 October 2015PUBLISHED
19 November 2015

Content from this work
may be used under the
terms of the [Creative
Commons Attribution 3.0
licence](#).

Any further distribution of
this work must maintain
attribution to the
author(s) and the title of
the work, journal citation
and DOI.



Abstract

Concordant computation is a circuit-based model of quantum computation for mixed states, that assumes that all correlations within the register are discord-free (i.e. the correlations are essentially classical) at every step of the computation. The question of whether concordant computation always admits efficient simulation by a classical computer was first considered by Eastin in arXiv:[quant-ph/1006.4402v1](#), where an answer in the affirmative was given for circuits consisting only of one- and two-qubit gates. Building on this work, we develop the theory of classical simulation of concordant computation. We present a new framework for understanding such computations, argue that a larger class of concordant computations admit efficient simulation, and provide alternative proofs for the main results of arXiv:[quant-ph/1006.4402v1](#) with an emphasis on the exactness of simulation which is crucial for this model. We include detailed analysis of the arithmetic complexity for solving equations in the simulation, as well as extensions to larger gates and qudits. We explore the limitations of our approach, and discuss the challenges faced in developing efficient classical simulation algorithms for all concordant computations.

1. Introduction

Understanding the hardness of simulating quantum computation on a classical computer is a central question in quantum-computing theory. Efforts to address this question are important to help identify candidates for quantum algorithms which outperform their classical counterparts. An essential aspect is to identify classes of quantum algorithms which *fail* to admit any speed-up compared to their classical counterparts, since this can give us insights into the aspects of quantum mechanics that might be responsible for any quantum computational speedup. However, there are still relatively few general results in this area. Often insight can be gained by the development of efficient simulation methods for certain families of quantum physical processes and quantum circuits.

A prominent example is the role of entanglement in unitary circuit-based quantum computation over pure states [1, 2]. For this model, exponential speedup with respect to classical computation requires that certain measures of entanglement scale with problem size. This indicates that entanglement plays an important role in pure-state circuit quantum computation. However, entanglement-scaling on its own does not provide a sufficient condition for a computational speedup. For example, highly entangling circuits using only gates from the Clifford group can be efficiently simulated via the Gottesman–Knill theorem [3]. To add further nuance, there exist models of universal quantum computation where certain entanglement measures may remain small and even tend to zero with growing computational size [4]. These results demonstrate that the role played by entanglement in pure-state computations is a subtle one.

Much less is known, on the other hand, about quantum computation over mixed states. For pure states, absence of entanglement (separability) implies a tensor-product state, and coincides with the absence of any correlation. However, for mixed states separability is a much weaker constraint than being uncorrelated, and the

correlations in such states can exhibit both classical and non-classical correlations. A long-standing question is whether general unitary circuits acting on separable states can be efficiently simulated classically [1] (i.e. assuming that the register remains separable at every stage of the computation). Since classically hard probability distributions can be sampled from simple quantum circuits [5] and linear optical networks [6], quantum-generated states may be hard to classically simulate even in the absence of entanglement. Indeed, classical N -bit probability distributions require exponentially many parameters for their descriptions, just like entangled pure states.

A well-studied model of mixed-state computation is the DQC1 or ‘one-clean-qubit’ model, which uses a (partially)-pure control qubit and a register of qubits prepared in the maximally mixed state [7]. In this model, the normalized trace of a unitary circuit may be well approximated by the average of measurements on the control at the output. The role of entanglement in DQC1 was studied in [8]. It was found that entanglement in the output state, quantified using multiplicative negativity across bipartite cuts, becomes a vanishingly small fraction of the maximum possible as the size of the register increases [8].

Later studies looked at the generation of discord in the output state of DQC1 [9, 10], looking specifically at the correlations between the control qubit and the entire register. For ‘typical’ unitaries, defined as unitaries sampled using the Haar measure, it was found that discord remains a fixed fraction of the maximum as the number of register qubits increases [9]. We remark, however, that the normalized trace of Haar-random unitaries converges to zero as the size of the matrix gets large, since the corresponding eigenvalues are uniformly distributed phases between 0 and 2π . Hence the output of such DQC1 computations is known in this limit, and nothing can be concluded about algorithmic speedup. Reference [10] also provided a condition for the generation of no discord at the output of DQC1. Nonetheless, the relationship between entanglement and discord in cases of the DQC1 model with apparent algorithmic speedup remains little understood.

Important progress in the study of the role of correlations in mixed-state computation was made by Eastin in [11], where *concordant computation* was defined and first analysed. A concordant state, sometimes called a ‘fully classical’ state, is defined as a state which is diagonal in the computational basis up to local-unitary transformation. A concordant computation is one which satisfies the promise that the state of the system remains concordant after each unitary gate in the computation. Concordant states are closely related to classical probability distributions, their only non-classical attribute being the local-unitary freedom of their density-matrix eigenbasis. Thus they can be characterized via a probability distribution and local unitaries.

Monte-Carlo simulation has been the basis for a number of methods for efficiently simulating physical processes and quantum circuits [12–15]. Using a Monte-Carlo method, Eastin presents a general procedure [11] for simulating concordant computations which samples the output statistics. He argues that it is an efficient algorithm on a classical computer when circuits are restricted to one- and two-qubit gates. However, there are examples of concordant computation that admit efficient simulation but which do not fall within Eastin’s results.

Consider the following example of a concordant circuit for DQC1 using gates of unrestricted size: the initial state, comprising a pure control qubit and N fully mixed register qubits, is $|+\rangle_1 \otimes (1/2^N) \mathbb{I}_{2 \dots N+1}$ (where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$) and the circuit consists of a series of controlled-gates $G_1 \dots G_t$ which are Hermitian and diagonal in the computational basis, for which $G_k: |0\rangle_1 |\mathbf{x}\rangle_{2 \dots N+1} \mapsto |0\rangle_1 |\mathbf{x}\rangle_{2 \dots N+1}$ and $G_k: |1\rangle_1 |\mathbf{x}\rangle_{2 \dots N+1} \mapsto (-1)^{f_k(\mathbf{x})} |1\rangle_1 |\mathbf{x}\rangle_{2 \dots N+1}$ (where $f_k(\mathbf{x})$ takes values 0 or 1). The output state is then $(1/2^N) \sum_{\mathbf{x}|f(\mathbf{x})=0} |+\rangle_1 \otimes |\mathbf{x}\rangle_{2 \dots N+1} + (1/2^N) \sum_{\mathbf{x}|f(\mathbf{x})=1} |-\rangle_1 \otimes |\mathbf{x}\rangle_{2 \dots N+1}$ where $f(\mathbf{x}) = \sum_{k=1}^t f_k(\mathbf{x}) \pmod{2}$, and the expectation value for measurements on qubit 1 in the $|\pm\rangle$ basis is the average value for $(-1)^{f(\mathbf{x})}$ over all bit strings \mathbf{x} . This computation admits a straightforward Monte-Carlo simulation by sampling measurement outcomes for pure-state trajectories given input states $|+\rangle_1 |\mathbf{x}\rangle_{2 \dots N+1}$ (where \mathbf{x} is a bitstring chosen uniformly at random).

In this paper, we develop new technical tools to understand concordant computation, and use these to extend and refine Eastin’s results. In particular, we prove explicitly that our simulation is exact, an essential aspect of this model, since the concordance of a state is not preserved under arbitrarily small perturbations [16].

We proceed as follows: section 2 provides an informal introduction to concordant computation and some key ideas for simulating it. This includes general features of the states involved and specific requirements for simulating computations. Section 3 revisits the central results of [11] using a new formalism. It provides self-contained proofs within our revised framework. Section 4 presents a new simulation procedure that bypasses a bottleneck when identifying symmetries of the system state, which is a critical part of the procedure of [11]. In section 5 we explain the limitations of our simulation procedure, before concluding in section 6 with some discussion of the prospects of efficient simulations for all concordant computation.

2. Overview

In this section, we provide an overview of the techniques that are developed in the rest of this paper. We start with the following formal definition for a state to be concordant:

Definition. A state ρ , with N qudit subsystems labeled by j , of arbitrary dimension d_j , is called concordant if every qudit possesses a complete set of orthogonal rank-1 projectors $\pi_{k_j}^{(j)}$ such that

$$\rho = \sum_{k_1, k_2, \dots, k_N} p(k_1, k_2, \dots, k_N) \pi_{k_1}^{(1)} \otimes \pi_{k_2}^{(2)} \dots \otimes \pi_{k_N}^{(N)}, \quad (1)$$

where $p(k_1, k_2, \dots, k_N)$ is a probability distribution.

Concordant states have the interpretation of being the only quantum states having zero non-classical correlation with respect to any bipartition of the subsystems (see for example [17]). (Non-classical correlations can be quantified using quantum discord [18] or a variety of related measures [19].) The basic premise of the model called concordant computation is that an algorithm is supplied, consisting of a choice of a concordant initial state, unitary circuit, and measurements, such that the quantum state remains concordant after each gate acts. The generation (or ‘encoding’) of algorithms is not of concern here—only the simulation (‘decoding’) of algorithms which satisfy the promise of concordant states between gates.

The most basic example of concordant computation is given by probabilistic classical computation using reversible gates—which amounts to concordant computation in the computational basis. The expectation values for observables at the output can be evaluated efficiently by a Monte-Carlo method, which uses simulation trajectories on bit (dit) strings which are computed directly from the circuit given for the computation. More generally, concordant computations can involve entangling gates and changes to the local basis as the computation proceeds. To illustrate, if a CNOT gate acts on a concordant state ρ , for which both qubits on the support of the gate are in the $|\pm\rangle$ basis, then these basis elements are permuted and ρ remains concordant. However, if the control qubit is in the $|\pm\rangle$ basis and the target qubit is in the computational basis, then the gate maps basis elements to Bell states, and ρ may or may not remain concordant depending on its symmetries: only if ρ is invariant under $|+0\rangle \leftrightarrow |-0\rangle$ and $|+1\rangle \leftrightarrow |-1\rangle$ is concordance preserved.

More generally, the symmetries of the quantum state play a central role in concordant computation. We say that a unitary S with support on qudits b is a symmetry of state ρ if $S\rho S^\dagger = \rho$. The collection of all such unitaries (on b) defines a symmetry (sub)group. For example, when ρ has fully non-degenerate eigenvalues, and is diagonal in the computational basis, the symmetries include the identity operators and any phase gate. To make the consequences of symmetry manifest, let us write concordant states in equation (1) in a different form. This form is related to the original definition of a concordant state as a state of zero discord—see section 3.1 for more details. Given any concordant state ρ , and any partition of the qubits (or qudits) into subsets a and b , ρ can always be written

$$\rho = \sum_k \tilde{\rho}_k^{(a)} \otimes \Pi_k^{(b)}, \quad (2)$$

where $\Pi_k^{(b)}$ is a set of orthogonal projectors which are related to the computational basis by local-unitary operations, and the $\tilde{\rho}_k^{(a)}$ are (un-normalized) density operators. After collecting terms in the sum for which $\tilde{\rho}_k^{(a)} = \tilde{\rho}_{k'}^{(a)}$, this decomposition of ρ becomes unique (as proved in section 3.1) and we call the $\Pi_k^{(b)}$ in this case *full-rank subsystem eigenprojectors* (FRASEs). Any symmetry S of ρ with support on b then satisfies $S\Pi_k^{(b)}S^\dagger = \Pi_k^{(b)}$ for the corresponding FRASEs.

When the spectrum of ρ is fully non-degenerate, any set of gates which implements a concordant computation must map product states to product states at every step, and efficient Monte-Carlo trajectory simulation is (trivially) possible. However, when concordant states have degenerate eigenvalues, the problem of classically simulating concordant computation becomes non-trivial and more interesting. The degeneracy allows the gates specified in the problem to generate trajectories which create entanglement but leave the state concordant. A computational basis representation of such a trajectory will require exponentially growing resources.

The degeneracy in the quantum state, and the symmetries which follow from it, therefore disrupt naïve trajectory simulation. Fortunately, the degeneracy itself gives rise to a new way to construct trajectories by providing for families of equivalent unitary gates that lead to the same output state. We will say that gates G and \tilde{G} , with support on qudits b , are equivalent with respect to concordant state ρ , if $G\rho G^\dagger = \tilde{G}\rho \tilde{G}^\dagger$. It is easily verified that this last equation is also equivalent to the existence of a symmetry S of ρ on b satisfying $\tilde{G} = GS$.

Hence, the challenge for efficient classical simulation of a concordant computation is to find circuits of equivalent gates which define trajectories with an efficient simulation (where the computational requirements of *all* steps in the procedure are accounted for).

A key insight of Eastin in [11] is that if a gate G acting on subset of qudits b maps a concordant state to a concordant state, then it is equivalent to the following sequence of gate operations which also act only on b : a local unitary, a classical reversible gate, and a second local unitary. (By classical-reversible gate we mean a gate which permutes logical basis states, e.g. a CNOT, NOT or TOFFOLI gate.) The action of the two local unitaries is to first rotate the local basis of the state into the computational basis, and then rotate it to the new local basis for the state. If this set of alternative gates is known then a Monte-Carlo simulation will proceed via product states, and the classical simulation will be efficient. For any concordant computation this set of gates always exists, and the challenge is to efficiently compute it.

One approach to finding this equivalent gate set would be to identify all symmetries of the state (on b) and then search over gates to identify those with the needed properties. However, the symmetry identification can involve exponentially big matrices, since it involves an exhaustive search over the full state ρ , leading again to inefficient simulation. Furthermore, any attempt to find trajectories that relies on testing on the whole quantum state (e.g. direct application of the criteria for classicality given in [20]) will typically suffer from exponentially scaling overheads. As noted by Eastin, one can show that the identification of symmetries of an initially uncorrelated state after a circuit of unitary gates (even a set of classical reversible gates) is NP hard in general. Furthermore it is widely believed that not even universal quantum computers can solve NP-complete problems in polynomial time.

Nonetheless, Eastin argued that all necessary symmetry identification can be performed efficiently for concordant computations comprising circuits of one and two-qubit gates, together with some restriction on the form of the initial state (see section 3.2 for an alternative proof of this). Note that, although a universal gate set can be obtained using only one-qubit and two-qubit gates [21], it does not follow that concordant computations comprising three-and-higher qubit gates always (efficiently) decompose into concordant computations with one-qubit and two-qubit gates. Furthermore, the question of when qudit-based concordant computation for $d > 2$ (qudits) admits efficient classical simulation has remained entirely open so far.

In this paper, we develop a new approach to simulating concordant computation which allows us, in many cases, to go beyond the limitations discussed above. This will be described in detail in sections 4 and 5. The key new idea is that it is often not necessary to acquire full knowledge of the symmetries of the quantum state in order to identify classically efficiently simulable trajectories. In practice, the strict requirements upon the gate to leave the state concordant will often allow suitable trajectories to be extracted from individual quantum gates alone, side-stepping the NP-hard bottleneck in Eastin's algorithm with a tractable analysis on individual unitary gates.

The heart of our algorithm is a sub-routine that we call the *local-basis finder* (LBF). In section 4.1 we provide a detailed analysis of the LBF, which aims to identify the local basis rotation which forms the first of the three unitary gates (local rotation, classical reversible gate, local rotation) which act equivalently to the unitary gate applied in the circuit. Knowing this local rotation, the other two gates needed for the efficient trajectory simulation can be efficiently derived.

We emphasize that a critical issue for the simulation of concordant computation is the effect of numerical errors, such as rounding errors. The set of concordant states has zero volume relative to the Hilbert space of all quantum states. Small perturbations on concordant states will generate non-classical correlations (discord) [16], and necessarily disrupt the state symmetries which Eastin's algorithm computes at every step. Thus simulation algorithms of this type have no tolerance to such errors, and they must therefore proceed via exact numerical calculation. We remark that even the new algorithms introduced in this paper, where computing state symmetries is not always necessarily, require an exact representation of the local basis of the state for their successful implementation.

While Eastin did not consider this issue in [11], we show that his approach can be adapted to exact arithmetic while remaining efficient. The adoption of exact arithmetic is a non-trivial and a key technical contribution of our work. Exact simulation means that one cannot use the approximate floating-point arithmetic typically used to approximate real or complex numbers in physics simulations. We achieve this by adopting a combination of exact integer arithmetic and exact arithmetic on algebraic numbers. While the latter is not typically efficient [22, 23], we show in section 4.2 that this computational cost is a fixed overhead and does not affect the scaling of our algorithm (and our exact version of Eastin's algorithm).

3. Structure of concordant states and equivalent circuits for simulating concordant computation

In this section, we revisit the key results of [11], using an alternative argument with some new techniques that clarify how the approach works. In section 3.1 we introduce new tools for understanding a notion of degeneracy which plays a central role in [11] and in the current work. Then, in section 3.2, we formally derive the general method for trajectory-based simulation of concordant computation using these tools. We also review the difficulties encountered in [11] which centre around a step in the simulation algorithm—termed ‘diagnosing the degeneracy’—which attempts to identity symmetries of the system state.

3.1. Subsystem-eigenprojector (SE) decomposition for quantum-classical and concordant states

We begin by introducing a key notion of classicality relevant for concordant computation. A state ρ , with subsystems labeled a and b , is said to be *classical with respect to b* if there exists a complete set of rank-one projectors $\{\pi_k^{(b)}\}$ on b such that $\rho = \sum_k \pi_k^{(b)} \rho \pi_k^{(b)}$, or equivalently $\rho = \sum_k p_k \rho_{|k}^{(a)} \otimes \pi_k^{(b)}$ where $\{p_k\}$ is a probability distribution. A state of this form is sometimes referred to as a *quantum-classical state*, and the $\rho_{|k}^{(a)}$ are sometimes denoted conditional density matrices [19].

Consider the un-normalized conditional density matrix

$$\tilde{\rho}_k^{(a)} = p_k \rho_{|k}^{(a)}. \quad (3)$$

This operator satisfies an equation reminiscent of an eigenvalue equation

$$\rho \pi_k^{(b)} = \tilde{\rho}_k^{(a)} \otimes \pi_k^{(b)}, \quad (4)$$

with $\tilde{\rho}_k^{(a)}$ playing the role of the eigenvalue and $\pi_k^{(b)}$ the role of the eigenprojector. We shall see that in fact these operators do satisfy many of the properties of eigenvalues and eigenvectors respectively, and thereby provide a generalization of them.

Definition. A *subsystem operator-valued eigenvalue* (SOVE) $\tilde{\rho}^{(a)}$ and corresponding SE $\pi^{(b)}$ is any pair of such operators that satisfy

$$\rho (\mathbb{1}^{(a)} \otimes \pi_k^{(b)}) = (\mathbb{1}^{(a)} \otimes \pi_k^{(b)}) \rho = \tilde{\rho}_k^{(a)} \otimes \pi_k^{(b)}, \quad (5)$$

where $\pi^{(b)}$ is assumed have rank one, $\tilde{\rho}^{(a)}$ has support solely on a and $\pi^{(b)}$ has support solely on b .

For a state ρ which is classical with respect to sub-system b , a SOVE can be computed from any corresponding SE via the equation $\tilde{\rho}_k^{(a)} = \text{Tr}_b[(\mathbb{1} \otimes \pi_k^{(b)}) \rho]$.

Lemma 3.1. Suppose that $\tilde{\rho}_1^{(a)}$ and $\tilde{\rho}_2^{(a)}$ are SOVEs with corresponding SEs $\pi_1^{(b)}$ and $\pi_2^{(b)}$. If the SOVEs are distinct then the SEs must be orthogonal.

Proof. The proof is identical to a well-known proof of the orthogonality of eigenprojectors with distinct eigenvalues, and we include it for completeness: We have $\rho (\mathbb{1}^{(a)} \otimes \pi_1^{(b)}) = \tilde{\rho}_1^{(a)} \otimes \pi_1^{(b)}$ and $(\mathbb{1}^{(a)} \otimes \pi_2^{(b)}) \rho = \tilde{\rho}_2^{(a)} \otimes \pi_2^{(b)}$. Hence, $\tilde{\rho}_1^{(a)} \otimes \text{Tr}(\pi_2^{(b)} \pi_1^{(b)}) = \tilde{\rho}_2^{(a)} \otimes \text{Tr}(\pi_2^{(b)} \pi_1^{(b)})$ but if $\tilde{\rho}_1^{(a)} \neq \tilde{\rho}_2^{(a)}$, then the only solution to this equation is $\text{Tr}(\pi_2^{(b)} \pi_1^{(b)}) = 0$.

For any given quantum-classical state $\rho = \sum_k \pi_k^{(b)} \rho \pi_k^{(b)}$, the SOVEs can be *degenerate*, i.e. there can be two (or more) rank-one projectors $\pi_k^{(b)}$ and $\pi_{k'}^{(b)}$ such that $p_k \rho_{|k}^{(a)} = p_{k'} \rho_{|k'}^{(a)}$. Note however that the decomposition of ρ here has a form reminiscent of a spectral decomposition of a Hermitian operator into eigenvalues and eigenprojectors. An elementary result is that sets of orthogonal rank-one eigenprojectors of Hermitian operators are not unique when the spectrum includes degenerate eigenvalues, and that uniqueness is recovered when rank-one eigenprojectors are combined into full-rank eigenprojectors, corresponding to maximal subsets of rank-one eigenprojectors for distinct eigenvalues.

Remark. For any finite-dimensional Hermitian operator ρ , there is a *unique* set of full-rank projectors Π_k such that, $\rho = \sum_k \text{Tr}(\rho \Pi_k) \Pi_k$, which also satisfy $\sum_k \Pi_k = \mathbb{1}$ and $\Pi_k \rho = \rho \Pi_k = \text{Tr}(\rho \Pi_k) \Pi_k$.

Here the uniqueness follows from the full-rank property, and the orthogonality of the eigenprojectors associated with different eigenvalues. Alternatively, it follows directly as a corollary of lemma 3.2 below.

Proceeding now by analogy, we make the following definitions:

Definition. Let ρ be quantum-classical state with respect to a bipartition into subsystems a and b . We define a FRASE for ρ to be any SE $\Pi_k^{(b)}$, with $\text{rank} \geq 1$, for which there does not exist any $\pi^{(b)}$ on b such that $\rho[\mathbb{1} \otimes (\Pi_k^{(b)} + \pi^{(b)})] = \tilde{\rho}_k^{(a)} \otimes (\Pi_k^{(b)} + \pi^{(b)})$, where $\tilde{\rho}_k^{(a)} = \text{Tr}_b[(\mathbb{1} \otimes \Pi_k^{(b)})\rho] / \text{Tr}_b(\Pi_k^{(b)})$. Then we call a decomposition of the form

$$\rho = \sum_k \tilde{\rho}_k^{(a)} \otimes \Pi_k^{(b)} \quad (6)$$

a FRASE decomposition, where $\Pi_k^{(b)}$ are orthogonal FRASEs satisfying $\sum_k \Pi_k^{(b)} = \mathbb{1}$, and every $\tilde{\rho}_k^{(a)}$ is a distinct Hermitian operator on a .

It is now straightforward to prove that FRASE decompositions can be made along similar lines to expansions of Hermitian operators in their eigenvalues and full-rank eigenprojectors:

Lemma 3.2. Every ρ which is quantum-classical with respect to a bipartition into subsystems a and b , possesses a unique FRASE decomposition as given by equation (6).

Proof. The existence of a FRASE decomposition for any quantum-classical state follows directly from the definition for these states given above (by combining SEs for degenerate SOVEs). The uniqueness follows immediately from the fact that each FRASE is a full-rank projector onto a subspace, and a full-rank projector onto a sub-space is unique.

FRASEs satisfy many similar properties to full-rank eigenprojectors, and the standard definition of eigenprojectors is recovered as b is extended to the whole system. The uniqueness of FRASE decomposition underpins the simulation methods in this paper.

Now lemma 3.2 above gives rise to the following corollary for concordant states, which provides a useful uniqueness argument which will be needed later on:

Corollary 3.3. Any subset b of qudits in a concordant state $\rho = \sum_{\mathbf{x}} L p(\mathbf{x})|\mathbf{x}\rangle\langle\mathbf{x}|L^\dagger$, where $\mathbf{x} = \{x^{(1)}, x^{(2)}, \dots\}$ labels the computational basis and $L = L^{(1)} \otimes L^{(2)} \dots$ denotes local-unitary rotations for every qudit, has a unique set of FRASEs $\{\Pi_k^{(b)}\}$ which possess a product basis (that is to say the FRASEs are a sum of orthogonal product states).

Proof. The restrictions of the components of ρ to the subset of qudits in b , $L^{(b)}|\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}|L^{(b)\dagger}$, provide an orthogonal set of SEs from which a unique set of FRASEs can be constructed following lemma 3.2.

It is important to note that the local basis for b itself may not be unique, as is the case for example when the SE decomposition yields one FRASE which is the maximally mixed state.

3.2. Monte-Carlo simulation of concordant computation

Now we formalize the central challenge tackled in this paper and [11]. Our notation is as follows: we are given unitary circuit C , consisting of unitary gates G_t for the t th time step, on a system of N qudits (with arbitrary dimensions). The partially completed unitary after the t th step is $U_t = \mathcal{T}\Pi_{k=1}^t G_k$ (where \mathcal{T} denotes that the product respects the temporal ordering of the unitaries). We are also given ρ_0 which must admit a polynomially sized description, and the promise that the state of the system at every step, $\rho_t = U_t \rho_0 U_t^\dagger$ (where ρ_0 is of the form $\rho_t = \sum_{\mathbf{x}} p_t(\mathbf{x}) L_t |\mathbf{x}\rangle\langle\mathbf{x}| L_t^\dagger$ and where L_0 and p_0 only are given as part of the specification of the problem). Then the overarching goal can be stated as: find an equivalent circuit C' , consisting of unitary gates G'_t with partial completion of the circuit $U'_t = \mathcal{T}\Pi_{k=1}^t G'_k$, such that $U_t \rho_0 U_t^\dagger = U'_t \rho_0 U'^{\dagger}_t$, but for which there are corresponding pure-state trajectories which are known to be efficiently simulable.

Using the results of section 3.1, we can now proceed to derive the general form of a circuit suitable for simulating C . At time step t , G_t defines a bipartition of the system qudits into its support b and the rest a . The system state after $t - 1$ has form

$$\rho_{t-1} = \sum_k \tilde{\rho}_k^{(a)} \otimes \Pi_k^{(b)}, \quad (7)$$

where $\tilde{\rho}_k^{(a)} = \text{Tr}_b[(\mathbb{1}^{(a)} \otimes \Pi_k^{(b)})\rho_{t-1}] / \text{Tr}_b(\Pi_k^{(b)})$ and $\{\Pi_k^{(b)}\}$ is the unique set of FRASEs following corollary 3.3, which are sums of orthogonal product states $L_{t-1}^{(b)} |\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}| L_{t-1}^{(b)\dagger}$. Then, at the end of time step t

$$\begin{aligned}\rho_t &= G_t \rho_{t-1} G_t^\dagger \\ &= \sum_k \tilde{\rho}_k^{(a)} \otimes G_t \Pi_k^{(b)} G_t^\dagger.\end{aligned}\quad (8)$$

By inspection, the operators $G_t \Pi_k^{(b)} G_t^\dagger$ are SEs of ρ_t for the same bipartition. Since the $\Pi_k^{(b)}$ are FRASEs for ρ_{t-1} , and G_t is unitary, it follows that the $G_t \Pi_k^{(b)} G_t^\dagger$ are FRASEs for ρ_t . Furthermore since ρ_t is concordant, the operators $G_t \Pi_k^{(b)} G_t^\dagger$ must be a sum of orthogonal product states $L_t^{(b)} |\mathbf{x}^{(b)}\rangle \langle \mathbf{x}^{(b)}| L_t^{(b)\dagger}$ by corollary 3.3. The uniqueness property of FRASEs now gives us the following key equation:

$$G_t \Pi_k^{(b)} G_t^\dagger = L_t^{(b)} D_t L_{t-1}^{(b)\dagger} \Pi_k^{(b)} L_{t-1}^{(b)} D_t^\dagger L_t^{(b)\dagger} \forall k, \quad (9)$$

where D_t accounts for the possibility of a permutation of the computational-basis states on b . (Note that there is typically freedom in choices for D_t .)

For the system state we have the equivalence

$$\rho_t = G_t \rho_{t-1} G_t^\dagger = L_t^{(b)} D_t L_{t-1}^{(b)\dagger} \rho_{t-1} L_{t-1}^{(b)} D_t^\dagger L_t^{(b)\dagger}. \quad (10)$$

From this equation it should be noted that L_t agrees with L_{t-1} other than (possibly) on the b , and that $p_t(\mathbf{x}) = p_{t-1}(\{\mathbf{y}, D_t^{-1}(\mathbf{z})\})$ (where \mathbf{z} is the part of \mathbf{x} in b , \mathbf{y} is the part of \mathbf{x} not in b and we write $D_t|\mathbf{z}\rangle = |D_t(\mathbf{z})\rangle$). For C as a whole we have

$$\begin{aligned}\rho_t &= U_t \rho_0 U_t^\dagger = \left\{ \mathcal{T} \Pi_{k=1}^t (L_k D_k L_{k-1}^\dagger) \right\} \rho_0 \left\{ \mathcal{T} \Pi_{k=1}^t (L_k D_k L_{k-1}^\dagger) \right\}^\dagger \\ &= L_t \left\{ \mathcal{T} \Pi_{k=1}^t D_k \right\} L_0^\dagger \rho_0 L_0 \left\{ \mathcal{T} \Pi_{k=1}^t D_k \right\}^\dagger L_t^\dagger.\end{aligned}\quad (11)$$

To find a complete simulation algorithm for concordant computation, the challenge now is to find the gates L_k and D_k (for all time steps) which make up C' , given the initial state and C . We will return to this challenge shortly however, and consider how the output statistics would be simulated supposing, for argument's sake, that C' has already been found. First, we observe that a simulation algorithm does not need to compute an explicit description of the full state at every time step, but can instead just record changes to the system state using an update rule in keeping with equation (10). A suitable update rule is: (i) record a new local basis, for every qudit in the support of the gate which acts, specified by a complete set of rank-1 projectors (which need not be unique); (ii) record a suitable permutation operator which acts on the support of the gate. Given such an update rule, the output statistics of C' would be sampled as follows:

Remark. Monte-Carlo simulation, using a pre-calculated update rule, of the output statistics for a given concordant computation with initial state $\rho_0 = \otimes_{j=1}^N L_0^{(j)} (p_0^{(j)}(0)|0\rangle\langle 0| + p_0^{(j)}(1)|1\rangle\langle 1| + \dots) L_0^{(j)\dagger}$:

- Define start of a stochastic trajectory, (in the computational basis), by randomly picking a N -digit qudit string s_{in} according to probability distributions $p_0^{(1)}(\cdot), \dots, p_0^{(N)}(\cdot)$.
- Permute $s_{\text{in}} \mapsto s_{\text{out}}$ using $\mathcal{T} \Pi_{k=1}^{t_f} D_k$ given by the update rule.
- Sample probabilities $\text{Tr}(|b^{(m_j)}\rangle\langle b^{(m_j)}| L_t^{(m_j)} |s_{\text{out}}^{(m_j)}\rangle\langle s_{\text{out}}^{(m_j)}| L_t^{(m_j)\dagger})$ on specified qudits m_j with measurement bases $\{|b^{(m_j)}\rangle\}$. This is equivalent to stochastically flipping some of the bits of s_{out} with probabilities defined by the measurement basis.
- Repeat, and gather statistics.

Note that the local-basis changes at intermediate steps are not required to define the trajectories—only the basis of the final measurements. However, the intermediate local-basis projectors do play an essential role elsewhere for deriving the update rule from the initial specification of the concordant computation (as tackled in section 4).

The simulation algorithm described in [11] outlines a procedure for finding C' from the specification of a concordant computation. For each time step t , the simulation algorithm works in three stages: The first stage is equivalent to finding the decomposition for ρ_{t-1} in equation (7) (termed ‘diagnosing the degeneracy’ in the reference), and is done by testing for the (permutation) symmetries of $L_0^\dagger \rho_0 L_0$ on the support of the partially completed circuit U_t . The second and third stages were not described in detail in the reference, but are loosely equivalent to solving our equation (9) for $L_t^{(b)}$ and then D_t . The first stage however constitutes an NP-hard problem in general (as explained in [11]), which undermines the success of the simulation algorithm.

One way around this is to place a restriction on C , so that ρ_0 only needs to be tested for a limited group of symmetries. Reference [11] made the stipulation that each G_k should have support on one or two qubits only, as in this case the test can be limited to the set of classical-reversible gates which are *linear* [21, 26]. Reference [11]

includes an efficient symmetry test for this case. Eastin's proof of efficiency is not written using standard quantum information techniques. To aid the reader, therefore, we present an alternative formulation of this result here. We present a theorem and corollary, of which the latter is equivalent to lemma 4 in [11].

As we show below, the efficiency of Eastin's test can be attributed to the fact that one- and two-qubit reversible classical gates (CNOT gates, NOT gates, and combinations) are in the Clifford group. After stating the more general theorem 3.4 we then derive corollary 3.5, equivalent to lemma 4 in [11].

Theorem 3.4. *Suppose that $L_0^\dagger \rho_0 L_0 = \otimes_{j=1}^N (L_0^{(j)\dagger} \rho_0^{(j)} L_0^{(j)})$ as above, which is factorized and diagonal in the computational basis, and that S_σ is Clifford unitary on the N qubits. Then: $S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger = L_0^\dagger \rho_0 L_0$ if and only if $\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger - L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\vee j)} \right] = 0 \quad \forall j$. The expectation values $\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger) Z^{(j)} \otimes \mathbb{1}^{(\vee j)} \right]$ and $\text{Tr} \left[(L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\vee j)} \right]$ can be computed efficiently, and the overall computational complexity for evaluating all the required expectation values scales quadratically with N .*

Proof. See appendix A.

Corollary 3.5. *FRASES can be computed efficiently for every step in a concordant computation on N qubits, for which the circuit C is composed entirely of one and two-qubit gates, and $L_0^\dagger \rho_0 L_0$ is factorized and diagonal in the computational basis.*

Proof. At time step t , it is necessary to find the FRASES for ρ_{t-1} on the support b of G_b , and is equivalent to finding the full symmetry group of ρ_{t-1} on b . The promise of concordant computation implies that $\rho_{t-1} = \sum \tilde{\rho}_k^{(a)} \otimes \Pi_k^{(b)}$ where the $\Pi_k^{(b)}$ are FRASEs, and it is required to find the projectors $L_{t-1}^\dagger \Pi_k^{(b)} L_{t-1}$ in the computational basis (L_{t-1} is known from the previous time step). This can be done by finding all classical reversible gates P on b satisfying, $(D_{t-1} \cdots D_1)^\dagger P (D_{t-1} \cdots D_1) (L_0^\dagger \rho_0 L_0) (D_1 \cdots D_{t-1})^\dagger P (D_1 \cdots D_{t-1}) = L_0^\dagger \rho_0 L_0$. Under the restriction to one and two-qubit gates, all of the classical reversible gates D_1, \dots, D_t and P are Clifford gates (since they can be generated using only NOT and CNOT gates). Theorem 3.4 can therefore be applied here, and it guarantees that all necessary symmetry tests can be performed efficiently.

4. New simulation algorithm for concordant computation with gates of arbitrary size

The aim of this section is to develop an algorithm which can be used to simulate concordant computations which involve gates of arbitrary size. We continue using the notation introduced in section 3.2 for states, gates and circuits. To recap from [11], the algorithm therein attempts to identity a FRASE decomposition at every time step (for the initial state ρ_0). As stated above, deriving the FRASE decomposition by considering the whole history of the computation and the symmetries of the input state is NP-hard, equivalent to testing satisfiability for an arbitrary Boolean function.

The NP-hardness is avoided in [11] by restricting the simulation to concordant circuits composed of gates with support on only one or two qubits. Reversible one- and two-qubit gates are all in the Clifford group, and the efficiency of this algorithm is stated as corollary 3.5. Here we develop an alternative approach.

The requirement of concordance places strong conditions on every quantum gate. In particular, the FRASE decomposition for the state after a gate G can often be derived from the properties of G alone. Here we develop an algorithm which exploits this. The algorithm produces output equivalent to the one in [11], namely a sequence of permutation gates D_t and local-projector changes $L_{t-1}^{(b)} |\mathbf{x}^{(b)}\rangle \langle \mathbf{x}^{(b)}| L_{t-1}^{(b)\dagger} \mapsto L_t^{(b)} |\mathbf{x}^{(b)}\rangle \langle \mathbf{x}^{(b)}| L_t^{(b)\dagger}$, which are used to construct Monte-Carlo trajectories as described in section 3.2.

In section 4.1, we first derive the general structure of any gate which satisfies the promise of having concordant states for the input and output, and then we will present a general method for solving for its projectors $\{L_t^{(b)} |\mathbf{x}^{(b)}\rangle \langle \mathbf{x}^{(b)}| L_t^{(b)\dagger}\}$ and D_t from the given unitary G_t (and previously derived $\{L_{t-1}^{(b)} |\mathbf{x}^{(b)}\rangle \langle \mathbf{x}^{(b)}| L_{t-1}^{(b)\dagger}\}$). In section 4.2, we explain how each of the steps in the LBF can be implemented using exact arithmetic, and review the computational resources required.

4.1. Local-basis-update equations and method of solution

We start with a technical remark necessary to characterize all gates occurring in concordant computations:

Remark. Suppose that \mathcal{B} is a partitioning of the labels of the computational-basis states for qudits in b . A unitary B is block diagonal with respect to the partitioning \mathcal{B} of the computational basis of b , which is to say that the matrix representation is block-diagonal up to (identical) reordering of the rows and columns, if and only if B commutes with all projectors $X_j = \sum_{\mathbf{x} \in \beta_j} |\mathbf{x}\rangle \langle \mathbf{x}|$ with $\beta_j \in \mathcal{B}$.

Proof.

$$\langle \mathbf{x} | [B, X_j] | \mathbf{x}' \rangle = \begin{cases} 0 & \text{if } \mathbf{x}, \mathbf{x}' \in \beta_j, \\ 0 & \text{if } \mathbf{x}, \mathbf{x}' \notin \beta_j, \\ \langle \mathbf{x} | B | \mathbf{x}' \rangle & \text{if } \mathbf{x} \notin \beta_j, \mathbf{x}' \in \beta_j, \\ -\langle \mathbf{x} | B | \mathbf{x}' \rangle & \text{if } \mathbf{x} \in \beta_j, \mathbf{x}' \notin \beta_j. \end{cases}$$

Following the line of argument in section 3.2, every gate specified in a given concordant computation can be decomposed as follows:

Lemma 4.1. *Each gate G_t specified for concordant circuit C at time step t can be decomposed as*

$$G_t = L_t^{(b)} D_t B_t L_{t-1}^{(b)\dagger}, \quad (12)$$

where $L_{t-1}^{(b)}$ is the local unitary on the support b of G_t after all previous time steps, D_t is a classical-reversible gate, and B_t is block diagonal, being a direct sum of components which act identically on the projectors $X_k = L_{t-1}^{(b)\dagger} \Pi_k^{(b)} L_{t-1}^{(b)}$, where $\{\Pi_k^{(b)}\}$ is the set of FRASEs for ρ_{t-1} .

Proof. This follows immediately from equation (9) writing

$$\left(D_t^\dagger L_t^{(b)\dagger} G_t L_{t-1}^{(b)} \right) \left(L_{t-1}^{(b)\dagger} \Pi_k^{(b)} L_{t-1}^{(b)} \right) = \left(L_{t-1}^{(b)\dagger} \Pi_k^{(b)} L_{t-1}^{(b)} \right) \left(D_t^\dagger L_t^{(b)\dagger} G_t L_{t-1}^{(b)} \right) \quad \forall k$$

using the remark above.

The LBF exploits the guarantee of a decomposition of G_t as by equation (12), to solve for $\{L_t^{(b)} | \mathbf{x}^{(b)} \rangle \langle \mathbf{x}^{(b)} | L_t^{(b)\dagger}\}$ and D_t . To do this, it runs over all possible computational-basis projectors X_k (of arbitrary rank) on the support b of G_t , and attempts to recover rank-one (pure) local-basis projectors by solving the following set of nonlinear equations (for each qudit j in b) to find unknown local basis projectors $\rho^{(j)}$,

$$\begin{aligned} \left[\mathbb{1}^{(b/j)} \otimes \rho^{(j)}, G_t L_{t-1}^{(b)} X_k L_{t-1}^{(b)\dagger} G_t^\dagger \right] &= 0 \quad (\text{i}) \\ \text{subject to,} \\ \text{Tr}(\rho^{(j)}) &= \text{Tr}(\rho^{(j)2}) = \text{Tr}(\rho^{(j)3}) = 1 \quad (\text{ii}). \end{aligned} \quad (13)$$

Solutions to these equations will be sums of local basis projectors that are mapped to local projectors by G . General solutions to (i), for each qudit, can be arbitrary linear combinations of the desired local projectors, and the constraints (ii) are required to solve for solutions that correspond to pure basis states. Specifically, constraints (ii) impose purity on general Hermitian operators. For qubits only the conditions on $\text{Tr}(\rho^{(j)})$ and $\text{Tr}(\rho^{(j)2})$ are required, whilst the additional condition on $\text{Tr}(\rho^{(j)3})$ is required when the dimension is greater than two [27].

Our method for solving equation (13) is as follows: first Gaussian elimination is used to find a general Hermitian solution $\rho^{(j)}$ for equation (13)(i). A random instance $\tilde{\rho}^{(j)}$ of $\rho^{(j)}$ typically has support on the same local-basis projectors as $\rho^{(j)}$. Hence to derive rank-one solutions of equation (13)(i) and (ii), a random choice is made for $\tilde{\rho}^{(j)}$, the eigenvalues of $\tilde{\rho}^{(j)}$ are found from its characteristic polynomial, and the corresponding eigenvector projectors are derived by back substitution into the eigenvector equation. (The process can be repeated to address rare cases where a bad choice is made for $\tilde{\rho}^{(j)}$.) For each X_k , there are three types of solution to equation (13):

- (1) A complete local basis cannot be found for every qudit of b (in which case $L_{t-1}^{(b)} X_k L_{t-1}^{(b)\dagger}$ does not correspond to a valid input FRASE).
- (2) A complete local basis of unique rank-one projectors is found for each qudit of b .
- (3) A complete local basis of rank-one projectors is found for each qudit of b , where at least some of the projectors are not unique. (This occurs when local-basis projectors are combined in $G_t L_{t-1}^{(b)} X_k L_{t-1}^{(b)\dagger} G_t^\dagger$ and there is an infinite family of solutions.)

Note that it is only the rank-one projectors, and not the local unitaries L_b that are needed for the simulation algorithm. In general the LBF finds a complete local basis only for a subset of the X_k , and we call this set χ . The occurrence of non-unique local projectors could potentially cause difficulties when comparing results for different choices of X_k . For example, when $X_k = \mathbb{1}$ any complete local basis on b is a solution to equation (13). To address this, we define the X_k -unique local basis (for each qudit of b) as the unique combinations of rank-one projectors of minimal rank which are common to all possible local-basis solutions of equation (13). In other

words, the projectors in a X_k -unique local basis are the smallest local-basis projectors which are uniquely determined by equation (13). We denote the X_k -unique local basis projectors by $\mathbb{1}^{(b/j)} \otimes \rho_u^{(j)}$, and they are easily found—for example by repeatedly solving for the local-basis. We denote by LB_k the full set of X_k -unique local-basis projectors whenever it is defined.

To implement a Monte-Carlo simulation of the concordant computation, as described in section 3.2, there must be an unambiguous update rule for every time step. An unambiguous update rule can be obtained for time step t , only if the LBF is able to identify a unique set of local-basis projectors compatible with all LB_k for $X_k \in \chi$. For this to be possible, it is necessary that the elements of each LB_k are common projector solutions for all $X_{k'} \in \chi$ —that is to say that $[\mathbb{1}^{(b/j)} \otimes \rho_u^{(j)}, G_t L_{t-1}^{(b)} X_{k'} L_{t-1}^{(b)\dagger} G_t^\dagger] = 0 \quad \forall \mathbb{1}^{(b/j)} \otimes \rho_u^{(j)} \in LB_k, X_{k'} \in \chi$ —and the LBF must test all these conditions. When these conditions are met we term the LB_k compatible, and the following lemma can be applied:

Lemma 4.2. *When the LB_k are compatible $\forall X_k \in \chi$, a complete set of rank-one local-basis projectors solutions can be constructed, $\{L_t | \mathbf{x}^{(b)} \rangle \langle \mathbf{x}^{(b)} | L_t^\dagger\}$, for which $[L_t | \mathbf{x}^{(b)} \rangle \langle \mathbf{x}^{(b)} | L_t^\dagger, G_t L_{t-1}^{(b)} X_k L_{t-1}^{(b)\dagger} G_t^\dagger] = 0 \quad \forall \mathbf{x}^{(b)}, X_k \in \chi$.*

Proof. The set $LB'_t = \left\{ \text{non-zero projectors } \mathbb{1}^{(b/j)} \otimes \left(\rho_{u_1}^{(j)} \rho_{u_2}^{(j)} \dots \rho_{u_{|x|}}^{(j)} \right) \mid \forall \mathbb{1}^{(b/j)} \otimes \rho_{u_k}^{(j)} \in LB_k, \forall j \right\}$. LB'_t is a complete local-basis projector set, and represents a fine-graining of all the LB_k , and $[\mathbb{1}^{(b/j)} \otimes \rho_u^{(j)}, G_t L_{t-1}^{(b)} X_{k'} L_{t-1}^{(b)\dagger} G_t^\dagger] = 0 \quad \forall \mathbb{1}^{(b/j)} \otimes \rho_u^{(j)} \in LB'_t, X_{k'} \in \chi$. To find a complete set of rank-one local-basis projectors solutions, the projectors $\rho_u^{(j)}$ where $\mathbb{1}^{(b/j)} \otimes \rho_u^{(j)} \in LB_t$ can be decomposed into rank-one projectors. The vectors that correspond to these rank-one projectors can be obtained by orthogonalizing the set of column-vector entries of $\rho_u^{(j)}$ using a Gram–Schmidt process; finally the projectors can be renormalized to have trace 1.

A summary of the key steps of our LBF routine in pseudo code is given in figure 1. The possibility and implications of gates having multiple inconsistent local-basis solutions is taken up in section 5.

4.2. Implementing the LBF using exact arithmetic

Errors in the local projectors from time step $t - 1$ in our simulation algorithm, $L_{t-1}^{(b)} | \mathbf{x}^{(b)} \rangle \langle \mathbf{x}^{(b)} | L_{t-1}^{(b)\dagger}$, can cause a failure to find a complete local basis for $G_t L_{t-1}^{(b)} X_k L_{t-1}^{(b)\dagger} G_t^\dagger$ for time step t , even though one must exist for the error-free case by the promise of concordant computation. The simulation algorithm proposed in section 4.1 has no way of detecting and correcting errors, and they must be prevented from occurring. To address this issue, we look in detail at implementation of our simulation algorithm using integer arithmetic. Important goals here are to avoid unreasonable restrictions on the form of the concordant computations which can be simulated using our algorithm, and to ensure that the LBF does not incur excessive demands on computational resources, which should scale polynomially with the number of time steps with reasonable constraints on number size and memory. We permit irrational numbers in our simulation algorithm when they can be handled using (integer-based) exact arithmetic, and we have found it necessary to involve computations on (irrational) algebraic numbers for some intermediate steps.

First we modify the definitions of concordant states and concordant computation used thus far. We call a gate or projector rational if its matrix representation in the computational basis has only rational entries. Augmenting the definition of a concordant state given in section 3.1, we define a concordant state ρ to be *rationally concordant* if every subsystem possesses a complete set of rational orthogonal rank-1 projectors $\pi_{k_j}^{(j)}$ such that $\rho = \sum_{k_1, k_2, \dots} p(k_1, k_2, \dots) \pi_{k_1}^{(1)} \otimes \pi_{k_2}^{(2)} \dots$, and $p(k_1, k_2, \dots)$ is a rational probability distribution. Then we can define a *rationally concordant* computation as a concordant computation for which the system states are also rationally concordant at every time step, and in addition the projectors and probability distributions defining the initial state are rational, as are the gates G_t for all time steps. As an aside, we point out that our choice to use projectors to represent local bases in our simulation algorithm, (rather than the matrices $L_t^{(j)}$ themselves), prevents many standard gates and states from being excluded by the definition of *rationally concordant* computation here. Our aim is to involve local rotations which are proportional to (complex)-integer matrices but (generically) have irrational (surd) normalization factors, such as the Hadamard gate. For projectors defined using (complex) integer or rational entries, normalization proceeds by dividing out the trace, and surds are avoided. However a gate such as the $\pi/8$ gate, which is $\begin{pmatrix} 1 & 0 \\ 0 & (1 + i)/\sqrt{2} \end{pmatrix}$, must be excluded whenever it would generate surd factors between entries of a local projector occurring in the simulation.

Lemma 4.3. *An implementation of the LBF described in section 4.1 using exact arithmetic finds a complete set of rational rank-one projectors for $\{L_t^{(b)} | \mathbf{x}^{(b)} \rangle \langle \mathbf{x}^{(b)} | L_t^{(b)\dagger}\}$, given rational G_t and rational rank-1 local basis projectors*

(i) Pseudo-code for LBF

1. INPUT Gate G_t with support on qudits in b and the set of rank-one projectors $L_{t-1}^{(b)}|\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}|L_{t-1}^{\dagger(b)}$
2. REPEAT For every projector X_k in the computational basis
 SUBROUTINE: SOLVE for local-basis projectors of $G_t L_{t-1} X_k L_{t-1}^\dagger G_t^\dagger$ (see (ii) below).
 IF complete local-basis solution THEN RECORD X_k -unique basis in LB_k and k in $EXISTS-LIST$
3. REPEAT for every k, k' in $EXISTS-LIST$
 IF all projectors in LB_k commute with $G_t L_{t-1} X_{k'} L_{t-1}^\dagger G_t^\dagger$ DO NOTHING
 ELSE RECORD “Incompatible solutions”
4. IF “Incompatible solutions” THEN OUTPUT “Local-basis ambiguity at time step t ” and STOP
 ELSE
 FIND Complete rank-one local-basis projector set compatible with LB_k for all k in $EXISTS-LIST$
 RECORD rank-one projectors in LB_t , and $B_t = \{\beta_j \mid \text{combining over } \mathbf{x}^{(b)} \in \beta_j \text{ defines unique projectors}\}$
 FIND (any) D_t consistent with $G_t L_{t-1} X_j L_{t-1}^\dagger G_t^\dagger = L_t D_t X_j D_t^\dagger L_t^\dagger \forall j$, where $X_j = \sum_{\mathbf{x}^{(b)} \in \beta_j} |\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}|$
5. OUTPUT LB_t and D_t

(ii) Pseudo-code for solving local-basis equations

1. REPEAT for each qudit j in (b) and for confirmation of the solution
 SET $\rho^{(j)} \propto c_0 \mathbb{1} + \sum_l c_l \sigma_l$ for Hermitian basis matrices σ_l and real expansion coefficients c_l
 SET $n_l = [\sigma_l, G_t L_{t-1} X_k L_{t-1}^\dagger G_t^\dagger]$
 SET linear-equation system Ξ as matrix equation $\sum_l c_l n_l = 0$
 APPLY gaussian elimination on Ξ for general Hermitian solution for $\rho^{(j)}$
 PICK random instance $\tilde{\rho}^{(j)}$ of $\rho^{(j)}$ by random choices for free c_l
 APPLY root solver on characteristic equation of $\tilde{\rho}^{(j)}$
 APPLY back substitution into eigenvalue equation for rank-one eigenprojectors $L_t |\mathbf{x}^{(j)}\rangle\langle\mathbf{x}^{(j)}| L_t^\dagger$ of $\tilde{\rho}^{(b)}$
2. IF complete local basis solution THEN FIND corresponding X_k -unique basis following Lemma IV.2

Figure 1. (i) Pseudo-code for LBF: uses promise of existence of a decomposition of gate G_t (following lemma 4.1), to find local projectors $\{L_t |\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}|\}$ and classical-reversible gate D_t on the support b of G_t . The pseudo-code heralds cases where G_t is consistent with multiple *incompatible* (complete) sets of local projectors. (ii) Pseudo-code for subroutine for solving nonlinear equation (13) for the local projectors. The subroutine can be repeated to exclude the possibility of pathological choices for $\tilde{\rho}^{(j)}$. When a complete local basis is found for all qudits in b , the subroutine returns the X_k -unique version of it; one way to find the X_k -unique basis is by using the promise of projectors with integer matrices (see section 4.2).

$\{L_{t-1}^{(b)} |\mathbf{x}^{(b)}\rangle\langle\mathbf{x}^{(b)}| L_{t-1}^{\dagger(b)}\}$, provided all possible local basis solutions for the gate are compatible. Computations using algebraic numbers can be required at intermediate steps.

Proof. Part (i) is for the LBF applied to a single projector, $G_t L_{t-1} X_k L_{t-1}^\dagger G_t^\dagger$. Part (ii) is for finding LB_t from the LB_k when they are compatible, and for resolving higher-rank projectors into rational rank-1 projectors.

- (i) We refer to figure 1(ii) for the steps involved in solving for X_k -unique local basis projectors, and we give an implementation for them using integer computations: by making integer choices for the Hermitian basis matrices σ_l , an integer system of equations Ξ can be obtained (for a specific qudit j). A Gaussian elimination method can be applied to Ξ to solve for the general integer Hermitian solution $\rho^{(j)}$. More specifically, the Hermite normal form for Ξ , (an analogue of reduced echelon form for matrices over the integers), can be obtained in polynomial time using Bareiss’s algorithm, without suffering an exponential blowup in the memory requirements (see chapter 10 of [22]). A random choice for (Hermitian) $\tilde{\rho}^{(j)}$ can be made which is an integer matrix (by integer choices of the free variables post Gaussian elimination). The characteristic equation for the eigenvalues of $\tilde{\rho}^{(j)}$ is then a (real) monic polynomial with integer coefficients, and its solutions must be real. The elementary rational root test for polynomials dictates that the roots are either integer or irrational algebraic numbers. Both integer and irrational roots play an essential role for finding local-basis projectors. Hence we note that exact arithmetic operations can be performed on algebraic numbers using only integer/rational computations. This can be done by manipulations of polynomials defining the algebraic numbers, for example using an encoding for which the polynomials are represented by companion matrices and field operations are performed using matrix manipulations (see [23] for an introductory discussion on this). A method based on Sturm’s theorem can be used to find the eigenvalues of $\tilde{\rho}^{(j)}$ (for a treatment of Sturm’s theorem see chapter 7 of [22]). This theorem can be applied to the characteristic equation for $\tilde{\rho}^{(j)}$ to find the number of distinct roots in any arbitrary interval $(I_1, I_2]$, by using a Sturm sequence for the characteristic polynomial. More specifically, the number of roots in the interval is given by the difference in the number

of sign changes for the values of Sturm sequence when evaluated at I_1 and I_2 . The eigenvalues of $\tilde{\rho}^{(i)}$ can be found by a simple search method which repeatedly bisects a starting interval, at each step selecting one half interval which contains at least one root. This search method finds the eigenvalues exactly when they are integer, and it generates an isolating interval when the eigenvalues are irrational.

Once the eigenvalues of $\tilde{\rho}^{(i)}$ have been found, back substitution is used to find the rank-1 projector solutions. These solutions must be renormalized to have trace value 1. Integer eigenvalues lead to rational eigenprojectors, which are already part of the required X_k -unique local-basis solution. The existence of rational local-basis projector solutions with rank greater leads to eigenvalues which are irrational, and the associated rank-1 eigenprojectors must also contain irrational numbers. It is necessary to test combinations of these rank-1 eigenprojectors to find higher-rank projectors which are rational overall. The minimal-rank rational projectors formed this way must be added to the X_k -unique local-basis solution. The promise of rational concordant computation guarantees that a complete rational local basis can be found for at least one X_k .

- (ii) When the LB_k are compatible, we can find a fined-grained complete local-basis projector set with elements of rank ≥ 1 following the proof of lemma 4.2 above. It is necessary to verify that higher-rank rational projectors can be decomposed into rank-one projectors which are also rational. For this we employ a modified form of Gram–Schmidt as follows: let v_1, v_2, \dots be the column vectors of projector $\rho_u^{(i)}$. The vectors v'_1, v'_2, \dots defined as

$$\begin{aligned} v'_1 &= v_1, \\ v'_2 &= \langle v'_1, v'_1 \rangle v_2 - \langle v_2, v'_1 \rangle^* v'_1, \\ v'_3 &= \langle v'_2, v'_2 \rangle \langle v'_1, v'_1 \rangle v_3 - \langle v_3, v'_2 \rangle^* \langle v'_1, v'_1 \rangle v'_2 - \langle v_3, v'_1 \rangle^* \langle v'_2, v'_2 \rangle v'_1, \\ v'_4 &= \langle v'_3, v'_3 \rangle \langle v'_2, v'_2 \rangle \langle v'_1, v'_1 \rangle v_4 - \langle v_4, v'_3 \rangle^* \langle v'_2, v'_2 \rangle \langle v'_1, v'_1 \rangle v'_3 \\ &\quad - \langle v_4, v'_2 \rangle^* \langle v'_3, v'_3 \rangle \langle v'_1, v'_1 \rangle v'_2 - \langle v_4, v'_1 \rangle^* \langle v'_3, v'_3 \rangle \langle v'_2, v'_2 \rangle v'_1 \\ &\quad \text{etc} \end{aligned}$$

are rational and orthogonal. The rank-one rational projectors

$|v_1\rangle\langle v_1|/\text{Tr}(|v_1\rangle\langle v_1|)$, $|v_2\rangle\langle v_2|/\text{Tr}(|v_2\rangle\langle v_2|)$, ... give the required decomposition into rank-one projectors.

Next we consider the computational requirements for the LBF implementation described in lemma 4.3 (for exact computation). First of all, we observe that the (worst-case) computational overhead for the LBF scales poorly with gate size: for a gate with support having dimension d , the total number of projectors of all ranks to which the LBF might be applied scales as $O(2^d)$, a scaling which is doubly exponential with respect to the number of qudits. In the worst case, the LBF finds a local-basis solution for only one pair of input projectors, and the LBF must be applied to input projectors of all rank between 1 and $[d/2]$ (note that Π and $\mathbb{1} - \Pi$ must share the same local basis). (In the simplest case, the LBF is applied first to all possible one-dimensional projectors and the output projectors are found to carry the same local-basis, in which case it not necessary to check higher-rank input projectors.) Hence when considering the computational complexity, we consider the dependence on circuit size for a fixed maximum gate size.

Focusing now on the complexity for computations performed by the LBF given a specific choice of input projector, we can see that all steps involved can be performed efficiently. The key mathematical steps used by the LBF are: Gaussian elimination (and back substitution), for which complexity scales polynomially with respect to the matrices involved, and root finding, which is efficient due to the use of a bisection method. Furthermore, the majority of the calculations performed by the LBF use only integer matrices. Where irrational numbers do occur, however, there are large computational overheads due to the need to perform arithmetic operations using algebraic numbers with no loss of precision. The promise of rational concordant computation ensures that all irrational contributions must cancel for the output. Consequently, the computational cost for handling algebraic numbers can be regarded as a fixed overhead that does not undermine the efficiency of the LBF, for increasing numbers of gates.

The computational requirements for the LBF are addressed by the following lemma:

Lemma 4.4. *The computational complexity to solve for local-basis updates following lemma 4.3 scales, in regard of both time and space (memory), polynomially with respect to the number of circuit gates (for a fixed maximum gate size), and the total bits required to represent each gate and the initial state of each qudit.*

Proof. See appendix B.

5. Concordant computation with gates which are consistent with incompatible choices for the local basis

The LBF cannot be successfully applied in all cases: it is possible that the output of the LBF for a given unitary is not unique. In this case, the LBF generates incompatible multiple solutions (where the notation of compatibility is laid out in section 4.1). This causes the simulation algorithm to fail, since an incorrect local basis may be chosen which would then cause the simulation algorithm to generate an entangling trajectory. A linear number of such events would lead to an exponential number of trajectories which would need to be tested, leading to an inefficient algorithm. Here, we explore some cases where this non-uniqueness arises. Note that the LBF will commonly output multiple outputs in ways which do not disrupt the algorithm, arising for example from reordering of the local-basis projectors. We are not interested in such cases here, since they are easy to identify and unproblematic, and we focus only on cases where the outputs are truly incompatible.

Where the LBF outputs such incompatible solutions, there is ambiguity for the corresponding local-basis update, and additional information is required to derive valid simulation trajectories. We have tested our LBF numerically, by applying it to gates of the form $G = LDBL'$ where L, D, B and L' were generated randomly in keeping with the general form laid out in lemma 4.1. In our numerical studies, we found that the LBF did not output incompatible solutions in a large number of cases. However, we have also found special cases where the LBF outputs incompatible solutions for local-basis projectors depending on how one-dimensional projectors for the input are combined, which we now illustrate using examples.

Our first special case is given by the gate $G_{\text{exc},1}$ which maps the computational basis to the basis of Bell states,

$$G_{\text{exc},1}: |j, k\rangle \mapsto Z^j \otimes X^k \left(\frac{|00\rangle + |11\rangle}{\sqrt{2}} \right). \text{ It is convenient to write the action of } G_{\text{exc},1} \text{ in the Pauli basis:}$$

$$\begin{aligned} G_{\text{exc},1} |00\rangle\langle 00| G_{\text{exc},1}^\dagger &= \frac{1}{4}(\mathbb{I} + X \otimes X - Y \otimes Y + Z \otimes Z), \\ G_{\text{exc},1} |01\rangle\langle 01| G_{\text{exc},1}^\dagger &= \frac{1}{4}(\mathbb{I} + X \otimes X + Y \otimes Y - Z \otimes Z), \\ G_{\text{exc},1} |10\rangle\langle 10| G_{\text{exc},1}^\dagger &= \frac{1}{4}(\mathbb{I} - X \otimes X + Y \otimes Y + Z \otimes Z), \\ G_{\text{exc},1} |11\rangle\langle 11| G_{\text{exc},1}^\dagger &= \frac{1}{4}(\mathbb{I} - X \otimes X - Y \otimes Y - Z \otimes Z). \end{aligned} \quad (14)$$

Noting that $|00\rangle\langle 00| + |11\rangle\langle 11| = \frac{1}{2}(\mathbb{I} + ZZ)$, and that by local rotations from the Z -basis to the X and Y -bases also $|++\rangle\langle ++| + |--\rangle\langle --| = \frac{1}{2}(\mathbb{I} + XX)$ and $|+i + i\rangle\langle +i + i| + |-i - i\rangle\langle -i - i| = \frac{1}{2}(\mathbb{I} + YY)$, we can see that $G_{\text{exc},1}$ maps rank-two projectors in the computational basis to rank-two FRASEs carrying either the X, Y or Z basis for both qubits. (Note that $G_{\text{exc},1}$ must assign the same local basis both to a projector and the difference of that projector with the identity.)

$G_{\text{exc},1}$ has a rather exceptional structure that exploits special features of the Bell states. In contrast, a generic class of gates which generate FRASEs carrying incompatible local-bases is provided by controlled local unitaries. A simple example for two qubits would be gate $G_{\text{exc},2} = cU$ which implements a rotation U on qubit 2 from the computation basis to any-other qubit basis, controlled by qubit 1. For the set of input FRASEs $\{|00\rangle\langle 00|, |01\rangle\langle 01|, |1\rangle\langle 1| \otimes \mathbb{I}\}$, $G_{\text{exc},2}$ outputs FRASEs with the computational basis for both qubits. For the set of input FRASEs $\{|10\rangle\langle 10|, |11\rangle\langle 11|, |0\rangle\langle 0| \otimes \mathbb{I}\}$, $G_{\text{exc},2}$ outputs FRASEs with the computational basis for qubit 1, and the rotated basis for qubit 2. Similar examples can be easily constructed for gates with support on arbitrary numbers of qudits, with arbitrary dimensions.

Our LBF routine heralds the occurrence of incompatible local-basis solutions, but is forced to stop in such cases in the absence of additional information concerning the correct solution to choose. One possibility is to consider restricted instances of concordant computations which use only gates which do not generate incompatible solutions. When however it is necessary to consider gates which generate incompatible solutions, it is clear that efficient heuristic tests will suffice to resolve local-basis ambiguities in many cases. Another approach is to apply the LBF to extended sequences of gates with the aim of finding a unique local-basis update overall. The efficiency of our simulation procedure however is only preserved when the local-basis ambiguity extends over gate sequences which scale logarithmically with respect to the number of circuit gates.

6. Conclusions

In this paper we contribute several new results on the problem of constructing efficient classical simulations for concordant computation. These new results include a method to solve for local-basis updates using exact

arithmetic, which we prove is efficient. However, our results fall short of a proof that all instances of concordant computation admit efficient simulation or, on the contrary, that this is impossible in principle. The fundamental difficulty for any simulation of concordant computation is the need to test properties for the full quantum state. These tests generically involve exponentially large matrices (other than in some special cases) and hence are inefficient. The simulation procedure of [11] involves symmetry tests on the full quantum state. It was proved by the author that these tests are computationally equivalent to solving 3-SAT, an NP-Complete problem, proving that the simulation cannot be efficient in general.

In this paper we have progressed beyond the simulation procedure in [11], by attempting to bypass exhaustive symmetry testing on the quantum state. Our alternative simulation procedure attempts to derive simulation trajectories directly from the circuit which is supplied in the problem. This approach can work as the concordant promise heavily constrains the structure of the gates that make up the circuit. Consequently our analysis has focused on our LBF subroutine, which is for deriving local-basis updates directly from the gates and input local-basis projectors. Our most important contribution is proof that local-basis updates can be computed efficiently using exact arithmetic. Furthermore, our investigations have uncovered two classes of special gates for which the LBF outputs multiple incompatible choices for the local-basis updates. In such cases additional information about the quantum state is required to derive valid simulation trajectories. We leave the problem of characterizing the full set of special gates as an open challenge. It is also important to determine if these gates can be used to generate examples of concordant computation that cannot be reduced to those in the class of probabilistic reversible classical computation.

An entirely different approach to overcoming the inefficient symmetry tests of [11] would be to replace NP-hard exact symmetry tests on the quantum state with efficient probabilistic sampling. Using ideas in [28], it is possible to devise efficient symmetry tests based on random sampling, where the probability for error is exponentially suppressed. However, this approach gives rise to two challenges. The first is to understand the effects of errors within the simulation, which have been circumvented in this paper by using exact methods. The second is to understand cases when probabilistic methods fundamentally cannot work. Algorithms are highly structured by nature, and typically they are not well modelled as random processes. It is possible that there are scenarios involving concordant computation which provably require knowledge of rare hard instances to achieve valid output statistics, where the hard instances foil probabilistic tests on the quantum state. We leave a full analysis of these issues as an open problem.

Acknowledgments

HC acknowledges funding and support for this work by the National Research Foundation and Ministry of Education, Singapore, as well as the University of Bristol. We thank Bryan Eastin for helpful discussions.

Appendix A. Proof of theorem 3.4

Theorem A.1. Suppose that $L_0^\dagger \rho_0 L_0 = \otimes_{j=1}^N (L_0^{(j)\dagger} \rho_0^{(j)} L_0^{(j)})$, which is factorized and diagonal in the computational basis, and that S_σ is Clifford unitary on the N qubits. Then: $S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger = L_0^\dagger \rho_0 L_0$ if and only if $\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger - L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] = 0 \quad \forall j$. The expectation values $\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right]$ and $\text{Tr} \left[(L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right]$ can be computed efficiently, and the overall computational complexity for evaluating all the required expectation values scales quadratically with N .

Proof. The forward direction is trivial. For the reverse, we assume that

$\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger - L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] = 0 \quad \forall j$. It is necessary to establish that S_σ leaves all products of Z and $\mathbb{1}$ operators unchanged. (Trivially this holds for the identity.) Since $L_0^\dagger \rho_0 L_0$ is diagonal in the computational basis, for each qubit j which is pure, $|0\rangle$ (or $|1\rangle$), $\text{Tr} \left[(L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right]$ and $\text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right]$ are 1 (or -1); furthermore, $|0\rangle$ (or $|1\rangle$) is the only possible qubit state which gives 1 (or -1). Hence equality of $S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger$ and $L_0^\dagger \rho_0 L_0$ is established for the pure qubits.

For the remaining qubits $L_0^{(j)\dagger} \rho_0^{(j)} L_0^{(j)} = \frac{1}{2}(\mathbb{1}^{(j)} + q_j Z^{(j)})$, and we denote the unique values for the q_j by Q_1, Q_2, \dots , where $1 > Q_1 > Q_2 > Q_3 \dots > -1$. We also denote corresponding subsets of qubits with $q_j = Q_k$ by $J(Q_k)$. The effect of S_σ on $L_0^\dagger \rho_0 L_0$ in the Pauli basis is to permute the expansion coefficients for $L_0^\dagger \rho_0 L_0$ amongst the basis elements.

We now consider basis elements corresponding to the largest expansion coefficient Q_1 in $L_0^\dagger \rho_0 L_0$ and $S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger$. For each

$j \in J(Q_1)$, $\text{Tr} \left[(L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] = \text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] \implies S_\sigma Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} S_\sigma^\dagger = Z^{(k)} \otimes \mathbb{1}^{(\setminus k)}$ where $q_k \in J(Q_1)$. An analogous statement also holds when there are multiple Z operators for multiple qubits of $J(Q_1)$, and so $S_\sigma \left[\bigotimes_{j \in J(Q_1)} \frac{1}{2} (\mathbb{1}^{(j)} + Q_1 Z^{(j)}) \right] \otimes \mathbb{1}^{(J(Q_1))} S_\sigma^\dagger = \left[\bigotimes_{j \in J(Q_1)} \frac{1}{2} (\mathbb{1}^{(j)} + Q_1 Z^{(j)}) \right] \otimes \mathbb{1}^{(J(Q_1))}$ also.

Proceeding now to Q_2 , a similar argument can be made as for Q_1 . The following now holds for each $j \in J(Q_2)$: $\text{Tr} \left[(L_0^\dagger \rho_0 L_0) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] = \text{Tr} \left[(S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger) Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} \right] \implies S_\sigma Z^{(j)} \otimes \mathbb{1}^{(\setminus j)} S_\sigma^\dagger = Z^{(k)} \otimes \mathbb{1}^{(\setminus k)}$ where $q_k \in J(Q_2)$: a possible ambiguity might be considered when the value of Q_2 coincides with a power of Q_1 , but in this case the corresponding basis element is already accounted for in the previous step. Hence it can be concluded that $S_\sigma \left[\bigotimes_{j \in J(Q_2)} \frac{1}{2} (\mathbb{1}^{(j)} + Q_2 Z^{(j)}) \right] \otimes \mathbb{1}^{(J(Q_2))} S_\sigma^\dagger = \left[\bigotimes_{j \in J(Q_2)} \frac{1}{2} (\mathbb{1}^{(j)} + Q_2 Z^{(j)}) \right] \otimes \mathbb{1}^{(J(Q_2))}$

The same argument can now be applied for Q_3 . Possible ambiguities which might be considered when the value of Q_3 coincides with a product of Q_1 's and Q_2 's can be discounted, because the corresponding basis elements are accounted for previously.

By induction it follows that $S_\sigma L_0^\dagger \rho_0 L_0 S_\sigma^\dagger = L_0^\dagger \rho_0 L_0$. Finally, all expectation values that must be computed for the theorem are of the form for a Pauli product operator acting on $L_0^\dagger \rho_0 L_0$, each of these can be computed (via Gottesman–Knill theorem [3] techniques) with linear complexity in N , there are $2N$ such expectation values to compute and therefore the overall scaling is quadratic.

Appendix B. Proof of lemma 4.4

Lemma B.1. *The computational complexity to solve for local-basis updates following lemma 4.3 scales, in regard of both time and space (memory), polynomially with respect to the number of circuit gates (for a fixed maximum gate size), and the total bits required to represent each gate and the initial state of each qudit.*

Proof. We will consider an arbitrary time step within the simulation, for which the LBF is applied to gate G_t and qudit j , with dimension d_j . The dimensions of the support of G_t is d . Estimates for the computational requirements of the different types of mathematical steps involved are as follows:

- *Arithmetic on integers:* The computation requirements for multiplication dominate over those for addition. The (time) complexity for multiplying two l -digit numbers scales as $O(l^2)$ and the output has $2l$ digits. For multiplication of matrices involving l -digit integer entries (for real and imaginary parts), the digit length for entries of the output matrix scales linearly with l , while the time complexity scales as third order in the matrix dimensions.
- *Gaussian elimination/back substitution:* Gaussian elimination and back substitution are used by the LBF first to find a solution $\tilde{p}^{(j)}$ to equation (13), and then to find eigenprojectors for $\tilde{p}^{(j)}$ (given specific eigenvalues). The complexity for Gaussian elimination (which has cubic scaling with respect to the number of unknowns) dominates that for back substitution (for which the scaling is quadratic). The linear system Ξ which must be solved to find $\tilde{p}^{(j)}$ is overdetermined, and requires Gaussian elimination on a matrix with dimensions $O(d^2)$ by $O(d_j^2)$. When Bareiss's algorithm is used, the number of elementary steps is $O(d^2 d_j^3)$ and the maximum number size is $O(d_j (\log d_j + l))$, where l is the maximum number of digits for the matrix entries at the start (see lecture 10 of [22]). Finding eigenprojectors for $\tilde{p}^{(j)}$ requires Gaussian elimination to a matrix with dimensions d_j by d_j , which is integer for the case of an integer eigenvalue, but which has contributions which are algebraic numbers when the eigenvalue is an algebraic number.
- *Root finding for integer polynomials of degree p :* Root finding must be used to find the eigenvalues of $\tilde{p}^{(j)}$ (for which roots must be found for its characteristic equation of degree $p = d_j$), and it is also used for arithmetic operations on algebraic numbers. Our method for root finding uses Sturm's theorem (see lecture 7 of [22]). Sturm's theorem states that the existence of roots within a given interval can be detected by evaluating a Sturm chain of $p + 1$ polynomials at both interval endpoints, and taking the difference of the number of sign changes for the chain. By testing for the existence of roots within each interval, the search region can be repeatedly subdivided to locate the roots. If l is the maximum digit-length of the polynomial coefficients, then the initial search region can be taken to be of size $O(2^l)$ (e.g. using Cauchy's bound for polynomial roots). For integer solutions the smallest search interval has length 2, and the number of bisections required to locate a root is $O(l)$. For irrational solutions, an additional running time $O(p^2 l^2)$ is sufficient to identify an isolating interval (see lecture 6 of [22]).
- *Arithmetic on algebraic numbers:* Gaussian elimination and back substitution must be performed on matrices mixing integers with irrational algebraic numbers when the eigenvalues of $\tilde{p}^{(j)}$ are irrational. One method for performing arithmetic on algebraic numbers is as follows: every algebraic number can be represented as an

integer polynomial which has the number as a root, together with an isolating interval (e.g. see lecture 6 of [22]). Arithmetic operations (i.e. addition, multiplication, number comparison, etc) can be done by performing simple computations on companion matrices associated with the polynomials (described for example in [23]), together with updates to isolating intervals using the bisection method described in above. There is a large overhead for executing these arithmetic operations due to the need for Kronecker (tensor) product operations on the companion matrices.

References

- [1] Jozsa R and Linden N 2003 *Proc. R. Soc. A* **459** 2011–32
- [2] Vidal G 2003 *Phys. Rev. Lett.* **91** 147902
- [3] Gottesman D 1999 *Group 22: Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics* ed S P Corney, R Delbourgo and P D Jarvis (Cambridge, MA: International) pp 32–43
- [4] Van den Nest M 2013 *Phys. Rev. Lett.* **110** 060504
- [5] Bremner M J, Jozsa R and Shepherd D J 2011 *Proc. R. Soc. A* **467** 459–72
Hoban M J et al 2014 *Phys. Rev. Lett.* **112** 140505
- [6] Aaronson S and Arkhipov A 2013 *Theor. Comput.* **9** 143–252
- [7] Knill E and Laflamme R 1998 *Phys. Rev. Lett.* **81** 5672–75
- [8] Datta A, Flammia S T and Caves C M 2005 *Phys. Rev. A* **72** 042316
- [9] Datta A, Shaji A and Caves C M 2008 *Phys. Rev. Lett.* **100** 050502
- [10] Dakić B, Vedral V and Brukner Č 2010 *Phys. Rev. Lett.* **105** 190502
- [11] Eastin B 2010 arXiv:quant-ph/1006.4402v1
- [12] Mari A and Eisert J 2012 *Phys. Rev. Lett.* **109** 230503
- [13] Veitch V, Ferrie C, Gross D and Emerson J 2012 *New J. Phys.* **14** 113011
- [14] Stahlke D 2014 *Phys. Rev. A* **90** 022302
- [15] Pashayan H, Wallman J J and Bartlett S D 2015 *Phys. Rev. Lett.* **115** 070501
- [16] Ferraro A, Aolita L, Cavalcanti D, Cucchietti F M and Acín A 2010 *Phys. Rev. A* **81** 052318
- [17] Oppenheim J, Horodecki M, Horodecki P and Horodecki R 2002 *Phys. Rev. Lett.* **89** 180402
Luo S 2008 *Phys. Rev. A* **77** 042303
- [18] Zurek W H 2000 *Ann. Phys. (Leipzig)* **9** 855–64
Henderson L and Vedral V 2001 *J. Phys. A: Math. Gen.* **34** 6899–905
- [19] Modi K, Brodutch A, Cable H, Paterek T and Vedral V 2012 *Rev. Mod. Phys.* **84** 1655–707
- [20] Chen L, Chitambar E, Modi K and Vacanti G 2011 *Phys. Rev. A* **83** 020101(R)
- [21] Nielsen M A and Chuang I L 2000 *Quantum Computation and Quantum Information (Cambridge Series on Information and the Natural Sciences)* (Cambridge: Cambridge University Press)
- [22] Yap C K 1999 *Fundamental Problems of Algorithmic Algebra* 1st edn (Oxford: Oxford University Press)
- [23] Li C-K and Lutzer D 2004 *Coll. Math. J.* **35** 307–09
- [24] Ollivier H and Zurek W H 2001 *Phys. Rev. Lett.* **88** 017901
- [25] Loehr N 2014 *Advanced Linear Algebra* (Boca Raton, FL: CRC Press) ch 7
- [26] Preskill J *Lecture Notes for Physics 219/Computer Science, 219* (<http://theory.caltech.edu/people/preskill/ph229/ch6>)
- [27] Jones N S and Linden N 2005 *Phys. Rev. A* **71** 012324
- [28] Van den Nest M 2011 *Quantum Inf. Comput.* **11** 784–812